Королев Борис, учебная практика, ИУ10-21

0) **VMware Workstation** — это программа, которая позволяет на одном компьютере запустить несколько операционных систем одновременно. Для этого она эмулирует отдельный компьютер и дает возможность осуществлять с ним те действия, которые можно совершать с обычным, а именно устанавливать операционные системы и приложения.

1) Загрузка дистрибутивов

а)

По ссылке (http://rutracker.org/forum/viewtopic.php?t=4943151) загружаем VMware Workstation.

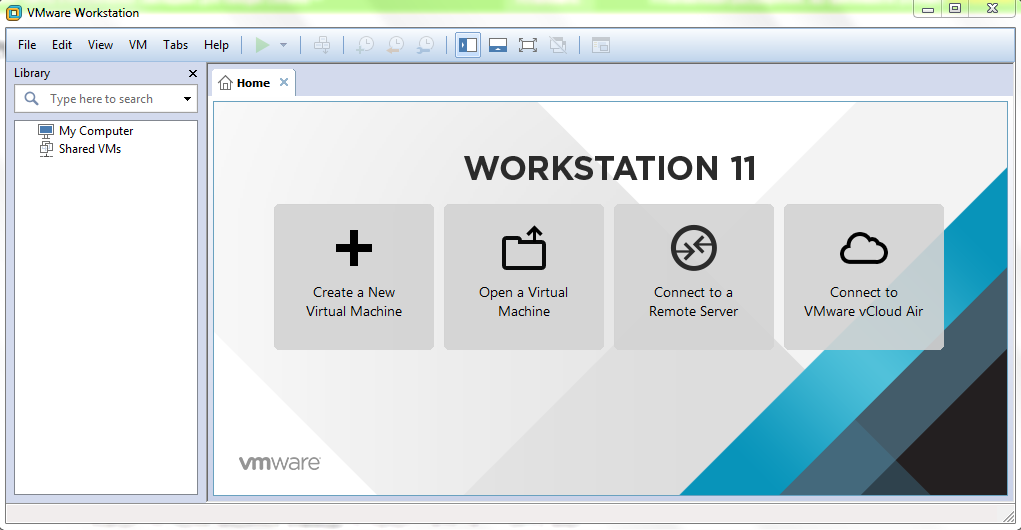
б)

Далее загружаем образ диска Centos по ссылке (http://mirror.logol.ru/centos/7/isos/x86\_64/CentOS-7-x86\_64-Minimal-1503-01.iso)

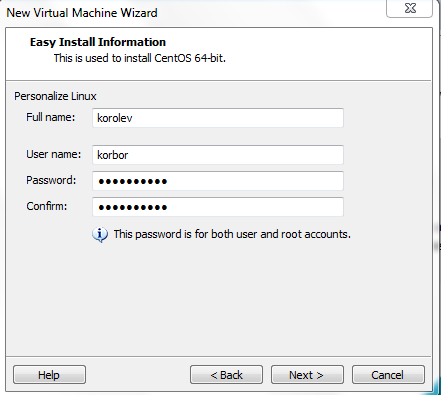
2) Установка WMware.

Открываем VMware Workstation после установки

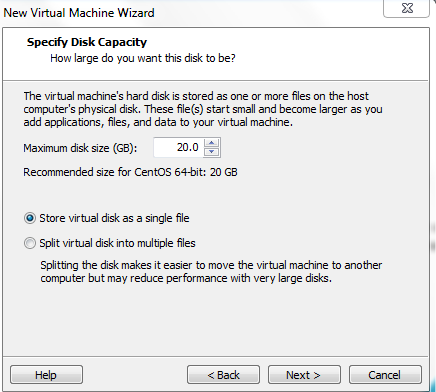
"Создаем новую виртуальную машину"



3) В ходе установки все пункты принимаем соглашение, а в этом создаем учетную запись, которая будет необходима для входа в систему.

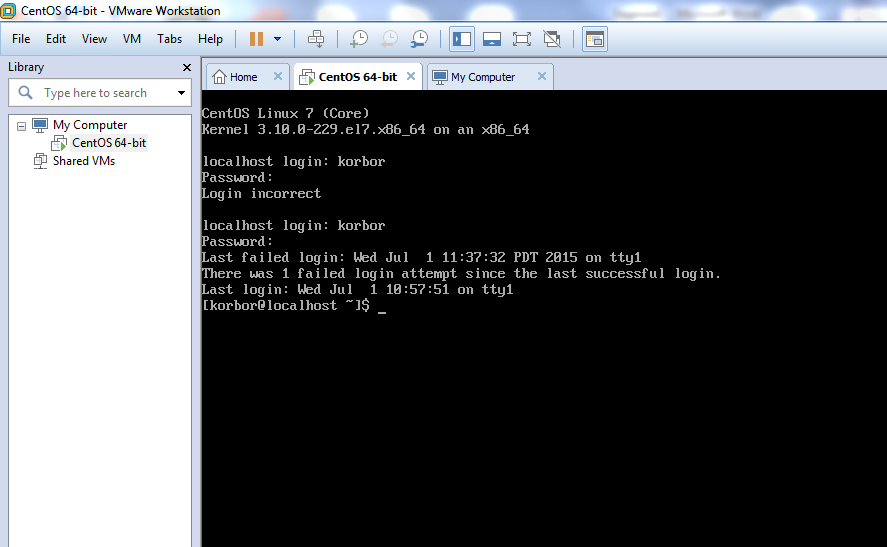


4) На данном этапе установке изменяем настройки на "Store virtual disk as a single file"



5) Вход в систему

На этом заключительном этапе заходим в свою учетную запись, если все было сделано верно, то система готова к использованию!



(в первом случае пароль введен неверно, во втором все верно)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2 Задание

Работа в консоли

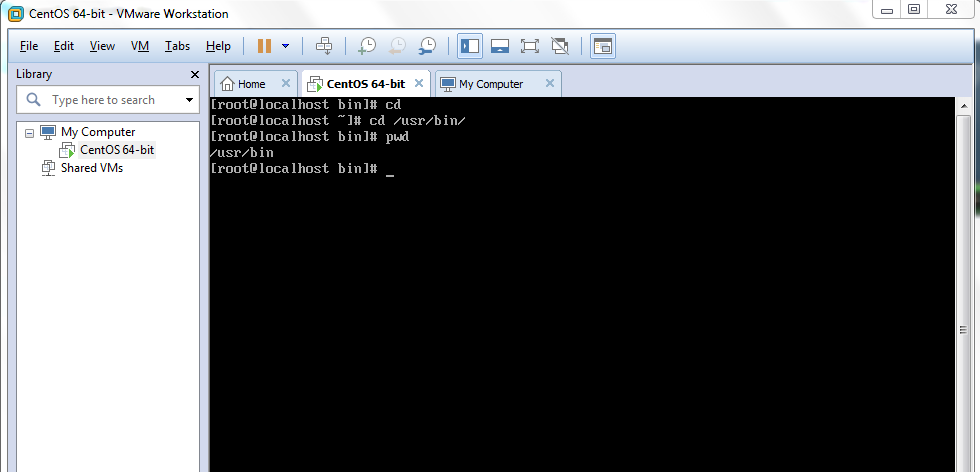
0) Командная строка – это текстовый интерфейс, в котором команды отдаются путем ввода текстовых строк с клавиатуры. Другое название командной строки, как Вы уже точно и давно поняли — консоль.

1) Команды

**cd**

Команда изменить текущий каталог — cd

Команда cd предназначена для смены каталога, в котором вы сейчас находитесь. Например, вы сейчас находитесь в своем домашнем каталоге ~ и хотите перейти в каталог ~/information, тогда нужно просто выполнить команду cd information.

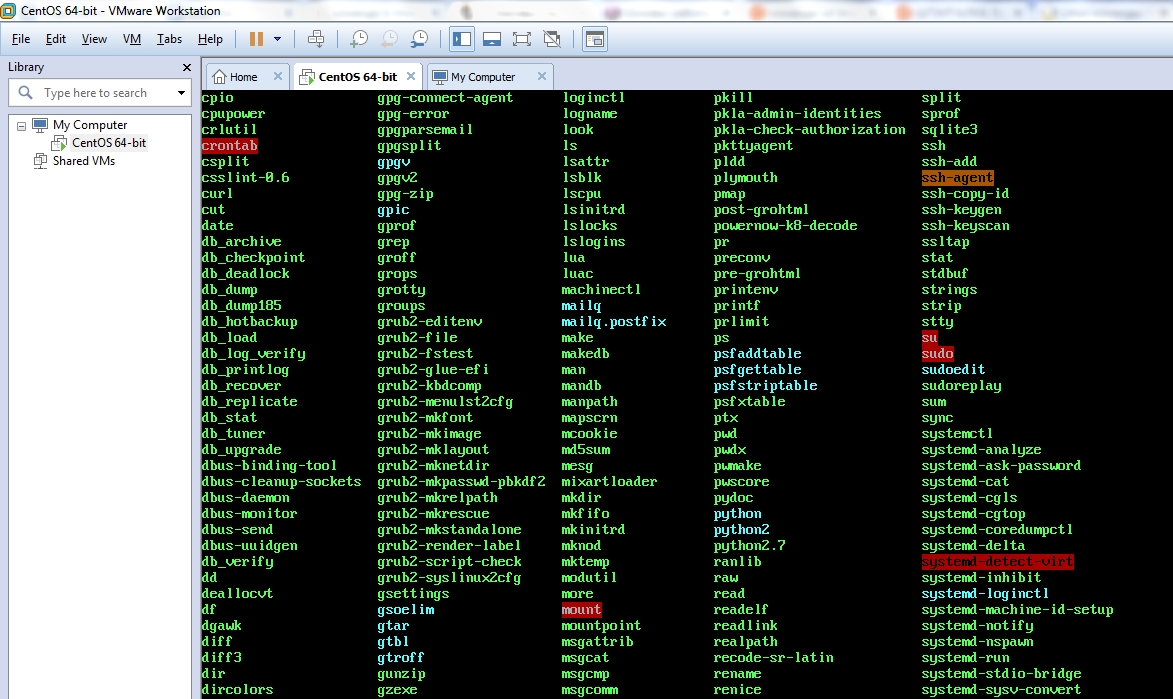
  
Вводим команду cd, далее переходим в папку bin, а с помощью команды pwd, проверяем в какой папке мы находимся.

**ls**

#### ls — вывод имен файлов

Утилита ls предназначена, для вывода списка файлов и каталогов в текущем каталоге. Рассмотрим некоторые полезные свойства команды ls.

-A — включить в список «скрытые» файлы («скрытые» файлы в Linux имеют в своем названии точку в начале имени файла, например, .gtkrc-2.0). -S — отсортировать файлы по размеру. -t — отсортировать по дате изменения. -1 — выводить имена в столбик, по одному на каждой строке. -l — вывести подробную информацию о каждом файле, размер файла, владелец файла, права доступа, дата изменения.

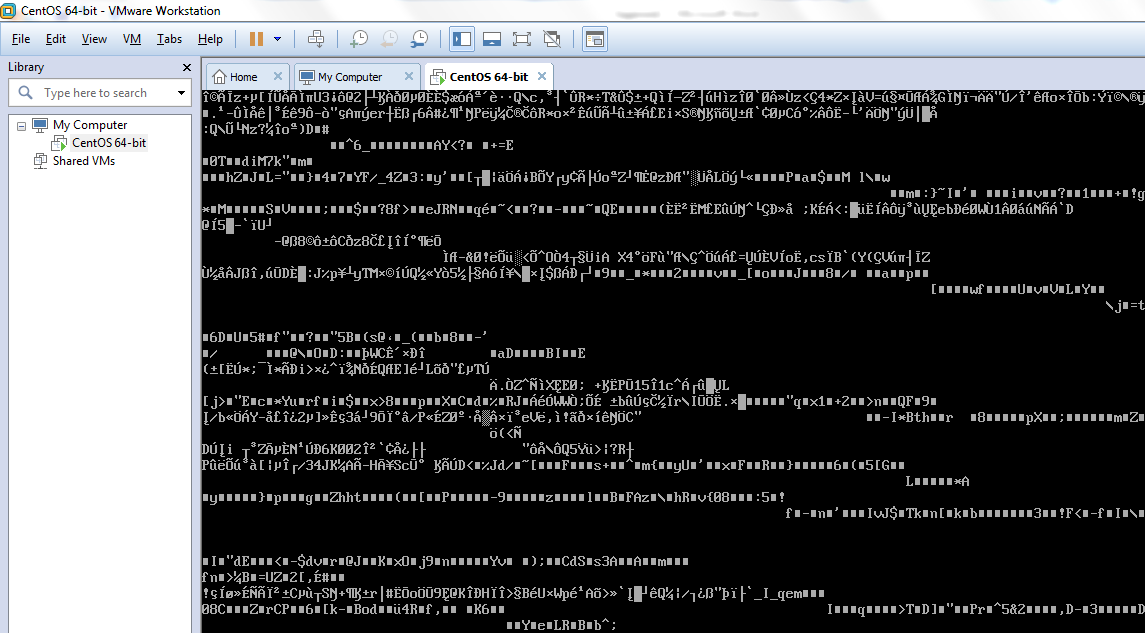


(Вывод всех файлов в папке)

**cat**

#### cat — вывод содержимого файла

Утилита cat выводит содержимое текстового файла. Простейший синтаксис команды cat <имя файла>. Утилита catможет использоваться для соединения конкатенации двух текстовых файлов, собственно от этого и происходит название утилиты (catenation, concatenation). Например, у нас есть два файла download и ubuntu и в каждом по две строки. Сначала мы выведем содержимое файлов по отдельности, а затем сделаем конкатенацию двух файлов.

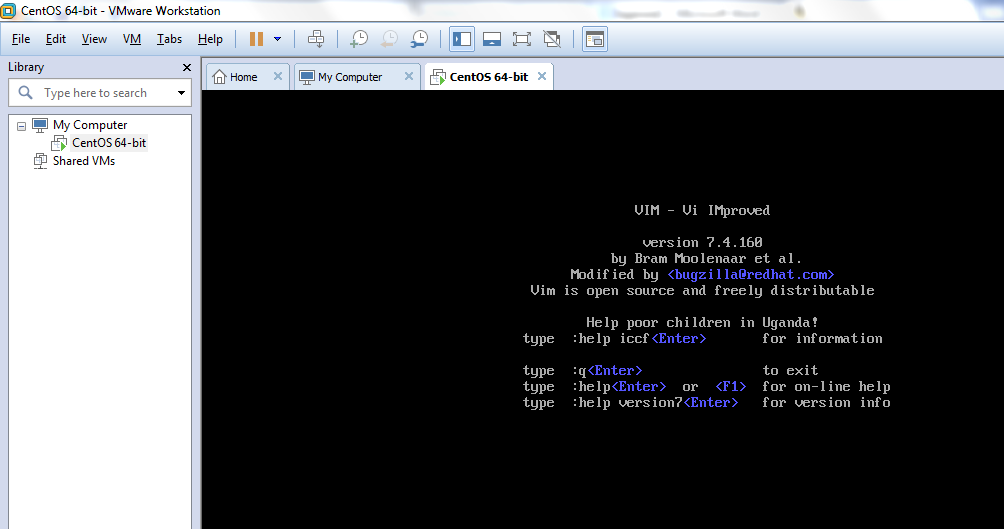


Отображение рандомного файла из папки bin с помощью функции cat.

**vi**

vi - активизирует экранно-ориентированный редактор

Редактор vi предлагает мощный набор операций для редактирования текста, основанный на некотором множестве мнемонических команд. Большинство команд - это одиночные клавиши, которые выполняют простые функции редактирования. Программа vi открывает "окно" размером с экран дисплея, в котором вы можете редактировать ваш файл. При помощи vi содержимое этого файла может быть легко и быстро изменено. При редактировании обеспечивается обратная визуальная связь (имя vi - сокращение от слова "visual").



10 самых нужных команд vi:

Режим команд:

1. **i** – Переход в режим редактирования перед текущим символом (вставка, insert).
2. **a**- Переход в режим редактирования после текущего символа (добавление, append).
3. **r** – Замена одного символа в текущей позиции курсора (replace).
4. **x**- Удаление символа в текущей позиции курсора.
5. **dd**- Удаление всей строки, в которой находиться курсор.
6. **u** – Отмена предыдущего изменения (undo).
7. **/шаблон**- Поиск шаблона в файле. Если шаблон встречается несколько раз, то можно использовать команды навигации: **n** – поиск вперед, **N** – поиск назад.
8. **yy** – Копирование текущей строки в буфер обмена (yank). Вставка – **p.**

Режим последней строки:

1. **:wq!**- Запись файла и выход из vi.
2. **:q! –**Выход из vi без записи текущих изменений.

## Вход с правами администратора root - режим "Super User" (su)

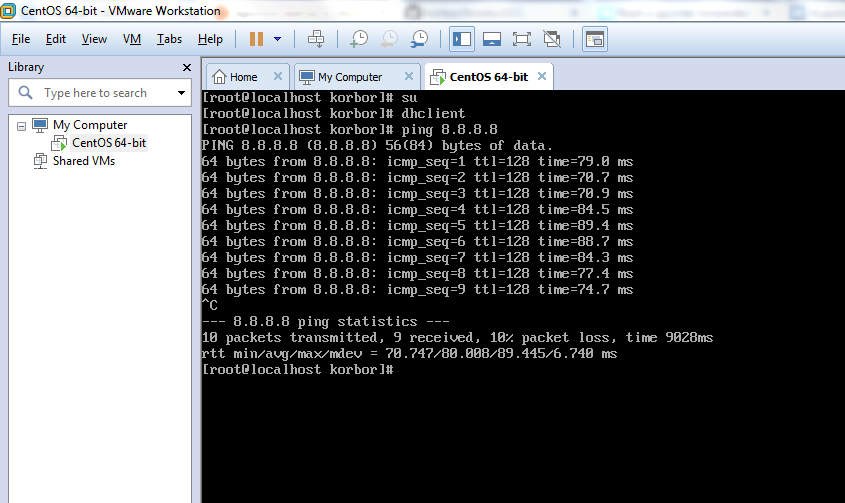
Команда su, отданная в консоли Linux вызовет запрос пароля администратора компьютера и, после его ввода, Ваш вход с правами администратора компьютера в той же консольной сессии. Это даёт возможность работать далее от имени администратора системы с любыми консольными командами и программами.

sudo — это утилита, предоставляющая привилегии root для выполнения административных операций в соответствии со своими настройками. Она позволяет легко контролировать доступ к важным приложениям в системе. По умолчанию, при установке Ubuntu первому пользователю (тому, который создаётся во время установки) предоставляются полные права на использование sudo.

**2) Настройка сети**

Чтобы настроить сеть, необходимо получить права рута, то есть вводим команду "su",

для подключения к сети вводим "dhclient", а дальше проверяем подключение к интернету, введя "ping 8.8.8.8"

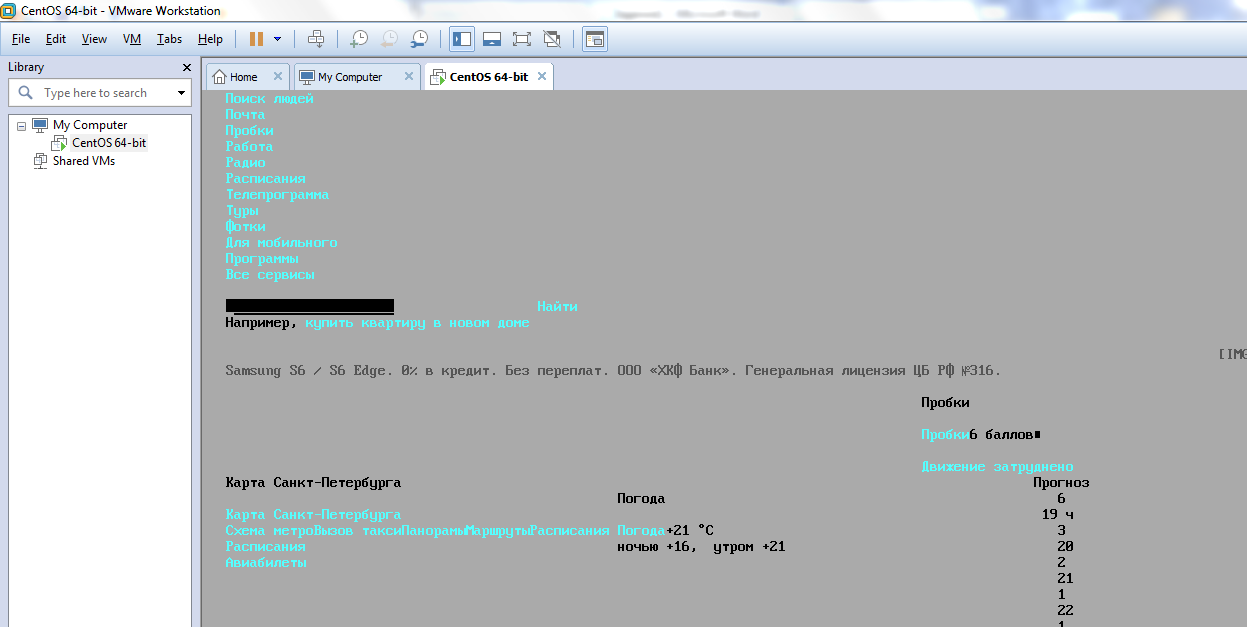


3) Установка дополнительных программ

Снова приобретаем права суперпользователя с помощью команды "su", далее вводим плагин для скачивания других прог "yum".

А) Установка браузера elinks

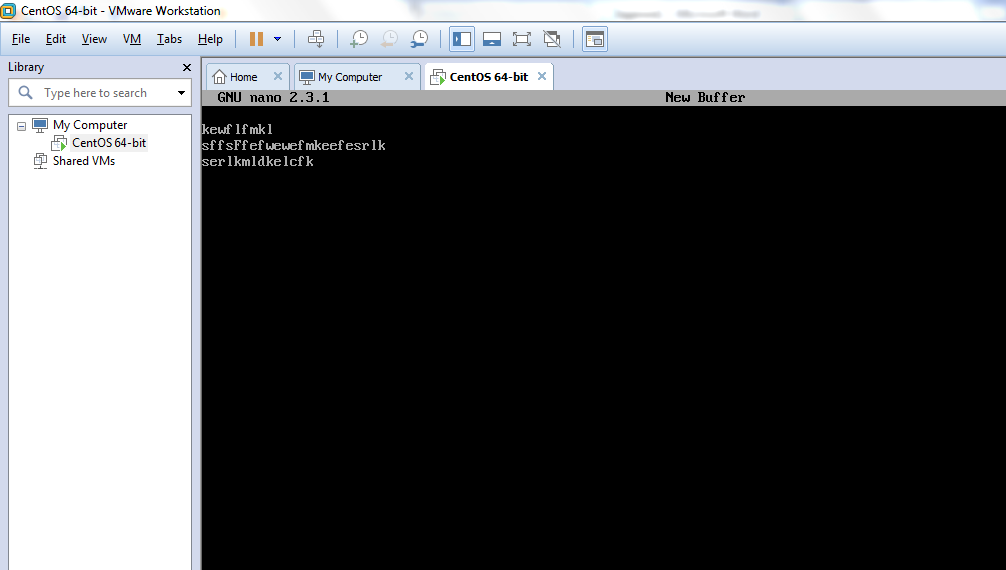
Для установки прописать в консоли:

"yum install elinks"

Б) Установка текстового редактора nano

Для установки прописать в консоли:

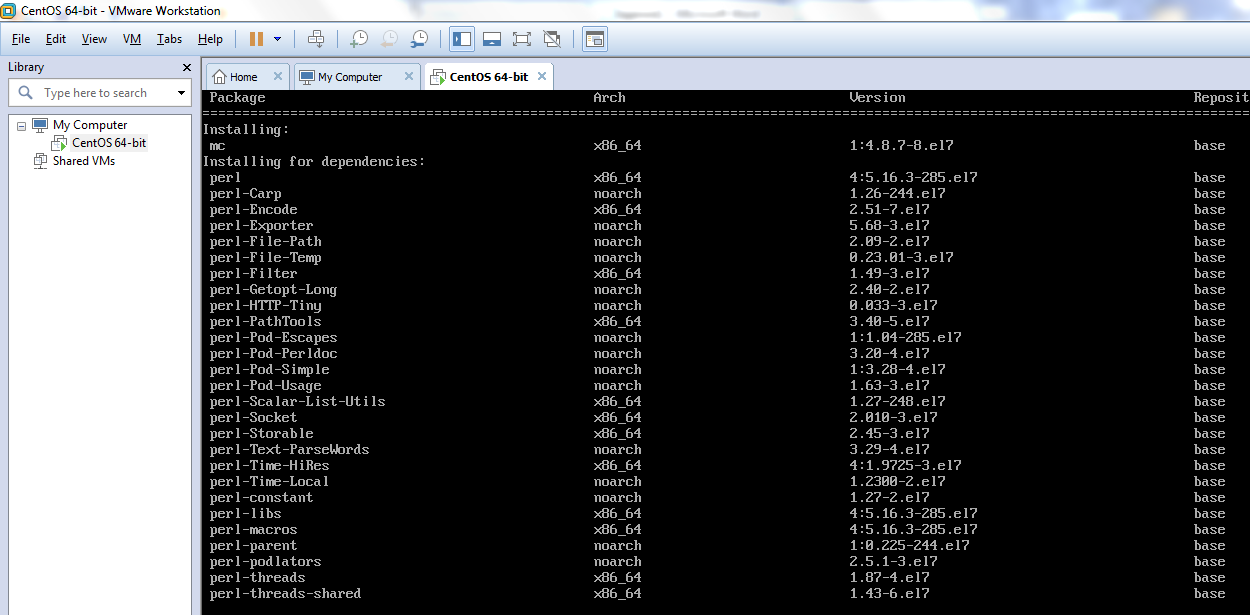
"yum install nano"



В) Установка файлового менеджера "midnight commander"

Для установки прописать в консоли:

"yum install mc"



**3 Задание**

**1) SSH-соединение**

Ssh (Secure Shell) - программа для регистрации на удаленной машине и выполнения команд на ней. Предназначена заменить rlogin и rsh, и обеспечить безопасную шифрованную связь между двумя компьютерам по незащищенной сети. Соединения X11 и произвольные TCP/IP порты могут также быть перенаправлены по защищенному каналу". Это мощная, очень удобная в работе программа, которая использует сильное шифрование для защиты всех передаваемых конфиденциальных данных, включая пароли.

В настоящее время существуют два SSH протокола, SSH2 и SSH1, первый является усовершенствованием SSH1. SSH2, помимо двойного шифрованного обмена ключами RSA, поддерживает и другие методы. Текущий дистрибутив поставляется со схемой обмена ключами Diffie-Hellman, имеет поддержку DSA и других алгоритмов открытых ключей.

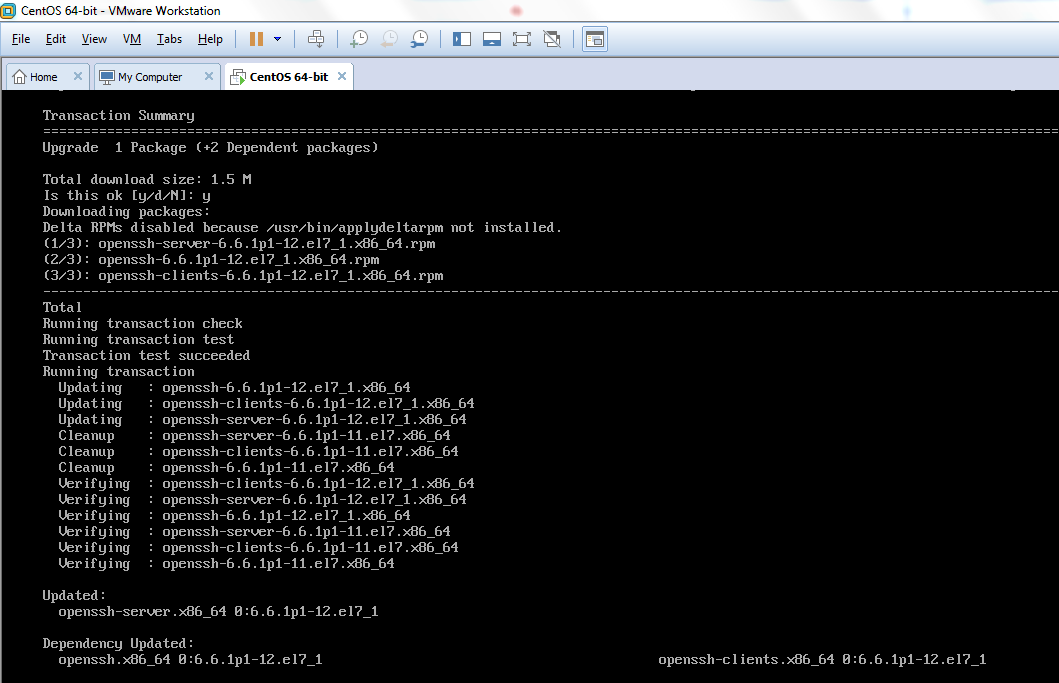
**Техническая информация**

SSH — это протокол прикладного уровня. SSH-сервер обычно прослушивает соединения на TCP-порту 22. Спецификация протокола SSH-2 содержится в [RFC 4251](https://tools.ietf.org/html/rfc4251). Для аутентификации сервера в SSH используется протокол аутентификации сторон на основе алгоритмов [электронно-цифровой подписи](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%86%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BF%D0%BE%D0%B4%D0%BF%D0%B8%D1%81%D1%8C) [RSA](https://ru.wikipedia.org/wiki/RSA) или [DSA](https://ru.wikipedia.org/wiki/DSA). Для аутентификации клиента также может использоваться ЭЦП RSA или DSA, но допускается также аутентификация при помощи пароля (режим обратной совместимости с[Telnet](https://ru.wikipedia.org/wiki/Telnet)) и даже ip-адреса хоста (режим обратной совместимости с [rlogin](https://ru.wikipedia.org/wiki/Rlogin)). Аутентификация по паролю наиболее распространена; она безопасна, так как пароль передаётся по зашифрованному виртуальному каналу. Аутентификация по ip-адресу небезопасна, эту возможность чаще всего отключают. Для создания общего секрета (сеансового ключа) используется [алгоритм Диффи — Хеллмана](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%94%D0%B8%D1%84%D1%84%D0%B8_%E2%80%94_%D0%A5%D0%B5%D0%BB%D0%BB%D0%BC%D0%B0%D0%BD%D0%B0) (DH). Для шифрования передаваемых данных используется [симметричное шифрование](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D0%BC%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%BE%D0%B5_%D1%88%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), алгоритмы [AES](https://ru.wikipedia.org/wiki/Advanced_Encryption_Standard), [Blowfish](https://ru.wikipedia.org/wiki/Blowfish) или [3DES](https://ru.wikipedia.org/wiki/3DES). Целостность передачи данных проверяется с помощью [CRC32](https://ru.wikipedia.org/wiki/CRC32) в SSH1 или [HMAC](https://ru.wikipedia.org/wiki/HMAC)-[SHA1](https://ru.wikipedia.org/wiki/SHA1)/[HMAC](https://ru.wikipedia.org/wiki/HMAC)-[MD5](https://ru.wikipedia.org/wiki/MD5) в SSH2.

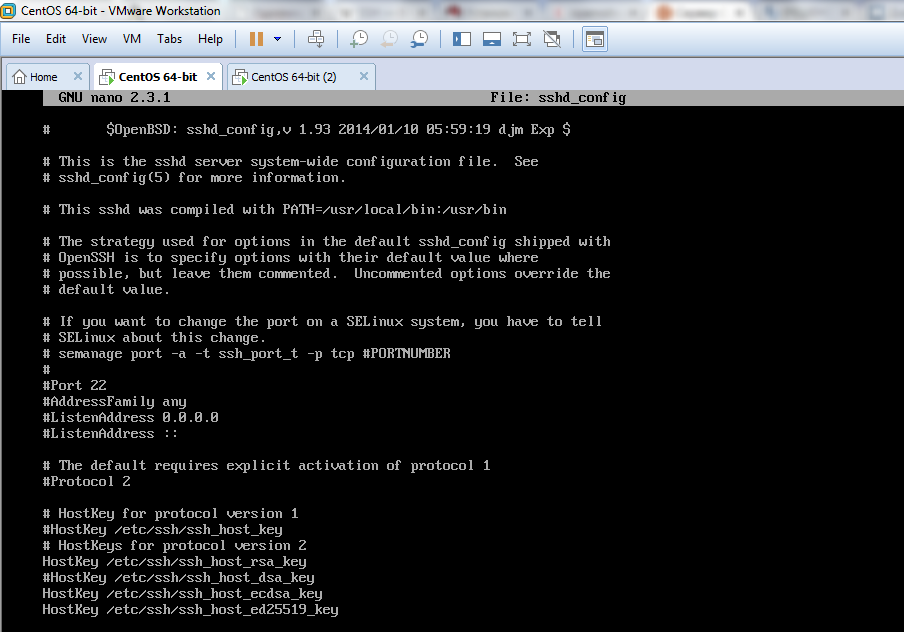
2) Установка Openssh-server

Для установки данной утилиты необходимо набрать простую команду в консоли, предварительно имея права рута

"yum install openssh-server"



Настройки sshd находятся в файле /etc/ssh/sshd\_config. Открываем этот файл для редактирования и изменяем его содержимое для наших нужд, не забывая при этом о безопасности.



Первый параметр – Port. По умолчанию используется 22 порт. Можно изменить его на нестандартный порт 2203 – это избавит наш сервер от сетевых роботов, которые автоматически сканируют интернет в поиске открытых портов и пытаются через них подключиться.

**#ListenAddress::**

**#ListenAddress 0.0.0.0 -** эти строки отвечают за настройку разграничений по сетевым интерфейсам, сетевому адресу или имени компьютера, по умолчанию сервер слушает (принимает подключения) на всех сетевых интерфейсах (оставляем без изменений)

Следующий параметр отвечает за версию протокола SSH. Значение по умолчанию 2. Его и оставляем. Не добавляем единичку ни при каких обстоятельствах. Первая версия протокола SSH не безопасна!

**Protocol 2**

Строка Privilege Separation указывает, должен ли ssh (sshd) разделять привилегии. Если да - то сначала будет создан непривилегированный дочерний процесс для входящего сетевого трафика. После успешной авторизации будет создан другой процесс с привилегиями вошедшего пользователя. Основная цель разделения привилегий - предотвращение превышения прав доступа. Оставляем yes.

**UsePrivilegeSeparation yes**

Далее следует группа параметров, отвечающая за журналирование. События, связанные с доступом по SSH записываются в журнал /var/log/auth

Первый параметр определяет, список каких событий администратор хочет видеть в журнале. Доступны следующие значения: DAEMON, USER, AUTH, LOCAL0, LOCAL1, LOCAL2, LOCAL3, LOCAL4, LOCAL5, LOCAL6, LOCAL7. Нас интересует авторизация, поэтому оставляем AUTH.

**SyslogFacility AUTH**

Второй параметр определяет уровень детализации событий. Доступны следующие события: SILENT , QUIET, FATAL, ERROR, INFO, VERBOSE, DEBUG, DEBUG1, DEBUG2, DEBUG3. Оставляем уровень детализации по умолчанию, если возникнут ошибки, то всегда можно увеличить детализацию журнала.

**LogLevel INFO**

Следующая группа параметров относится к аутентификации. Первый параметр означает, что соединение будет разорвано через указанное количество секунд, если пользователь не войдёт в систему. Уменьшим это время в два раза.

**LoginGraceTime 60**

Второй параметр разрешает или запрещает вход по SSH под суперпользователем. Запрещаем вход суперпользователю.

**PermitRootLogin no**

Третий параметр включает проверку демоном ssh прав и владение домашним каталогом пользователя, который пытается получить удалённый доступ к компьютеру. Оставляем yes.

**StrictModes yes**

Добавляем параметр AllowUsers, которого нет в конфигурационном файле по умолчанию. Этот параметр разрешает доступ к серверу по протоколу SSH только для перечисленных пользователей.

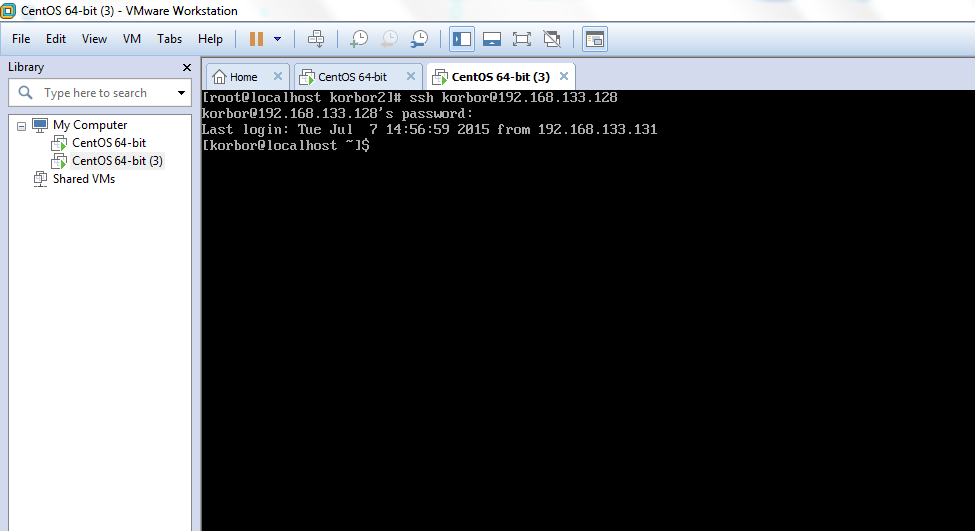
**AllowUsers korbor2**

3) Создаем вторую виртуальную машину, все настройки в первом пункте по аналогии.

Узнаем айпи первой машины с помощью команды "ip a"

теперь с помощью второй системы соединяемся первой машиной с помощью команды

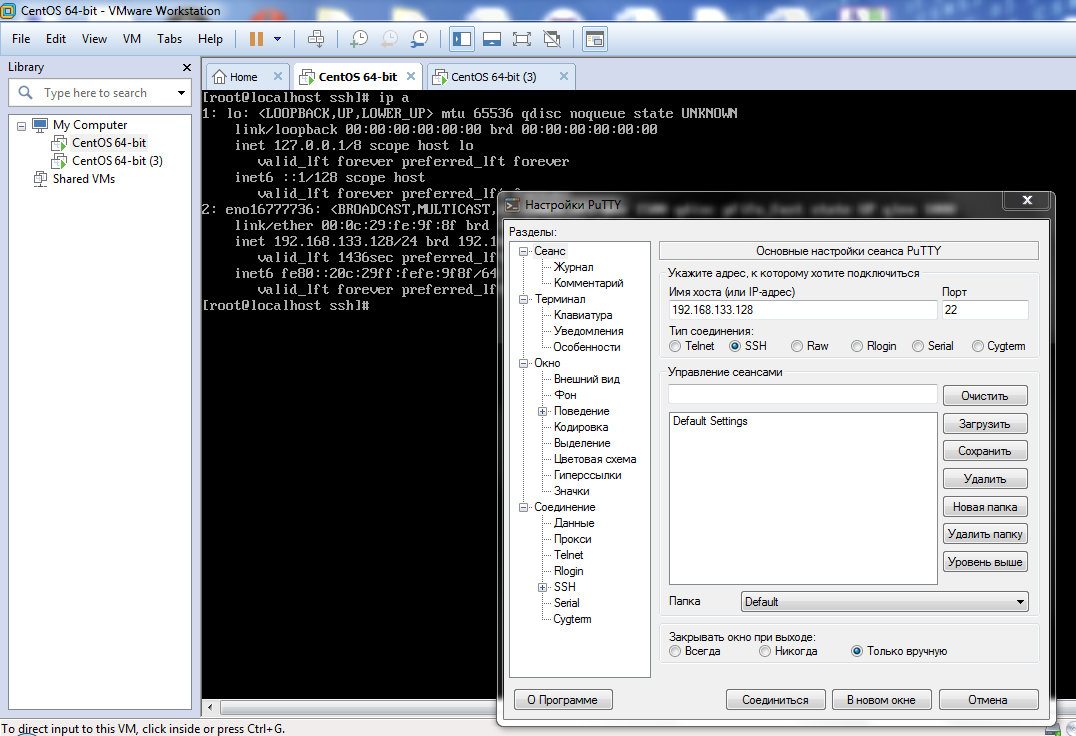
*ssh имя\_удалённого\_сервера@айпи\_удалённого\_сервера*



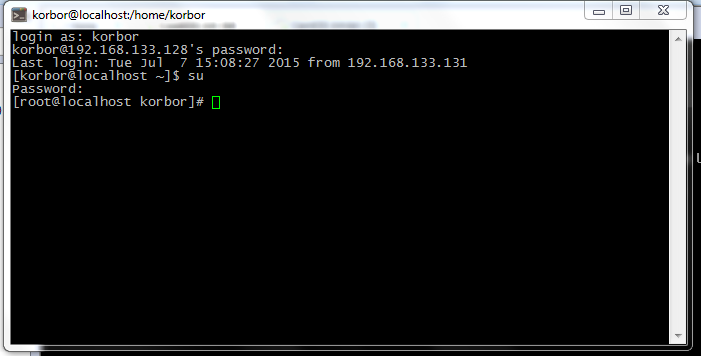
Соединение установлено, мы зашли со второй машины на первую.

скачиваем putty (*http://www.putty.org/ )*

запускаем putty, вводим айпи адрес виртуальной машины, на которую мы хотим зайти

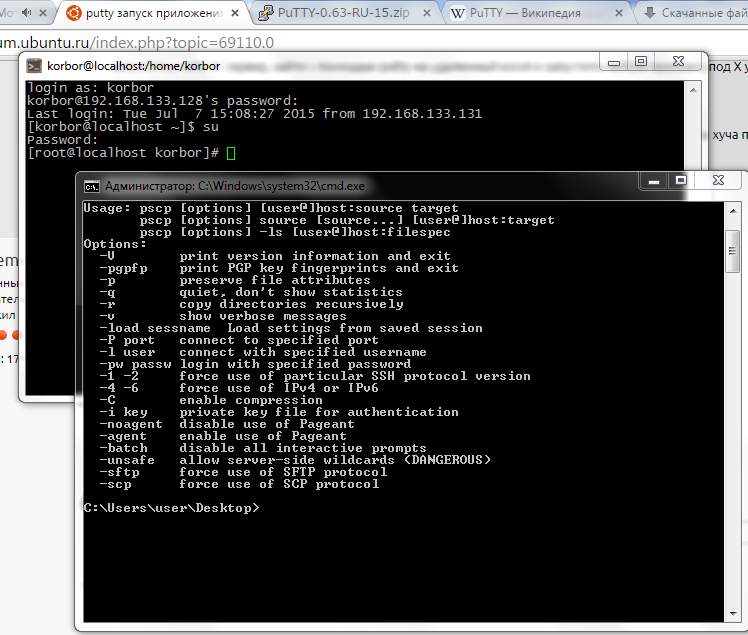


вводим логин и пароль от виртуальной машины



доступ получен.

при установке putty, в архиве будет находится файл pscp.exe, его надо запустить через командную строку windows.



если мы хотим скопировать файл с реальной машины на виртуальную, пользуемся командой *pscp С:\\адресс файла (пробел) имя\_удалённого\_хоста@айпи\_удалённого\_хоста:/папка в которой будет сохренён файл*

для копирования же файла с ВМ просто *пишем pscp имя\_удалённого\_хоста@айпи\_удалённого\_хоста:/папка в которой харанится файл*

**5) SSH-ключи**

Ключи SSH служат средством идентификации вас при подключении к серверу SSH с использованием [**криптосистемы с открытым ключом**](http://en.wikipedia.org/wiki/ru:%D0%9A%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81_%D0%BE%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%BC_%D0%BA%D0%BB%D1%8E%D1%87%D0%BE%D0%BC) и [**аутентификации вызов-ответ**](http://en.wikipedia.org/wiki/ru:%D0%92%D1%8B%D0%B7%D0%BE%D0%B2-%D0%BE%D1%82%D0%B2%D0%B5%D1%82_(%D0%B0%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F)). Одним из непосредственных достоинств этого метода перед традиционной идентификацией с помощью пароля является то, что вы можете быть авторизованы на сервере без регулярной необходимости отсылать ваш пароль через сеть. Даже если кто-либо будет прослушивать ваше соединение, у него не будет возможности перехватить и взломать ваш пароль, поскольку фактически он никогда не передается. Также использование для идентификации ключей SSH устраняет риск, связанный с брут-форс (brute-force) атаками, за счет существенного уменьшения шанса атакующего угадать правильные учетные данные.

Идентификация при помощи ключей SSH предоставляет дополнительную безопасность, а также может быть более удобным способом, чем традиционная идентификация при помощи пароля. При использовании вместе с программой, называемой агентом SSH, ключи SSH могут подключать вас к серверу или нескольким серверам без необходимости помнить и вводить ваш пароль отдельно для каждой системы.

Ключи SSH являются парными: один из них - закрытый, другой - открытый. Закрытый ключ известен только вам, и он должен быть в безопасности. С другой стороны, открытый ключ может свободно раздаваться с любого сервера SSH, к которому вы хотите подключиться.

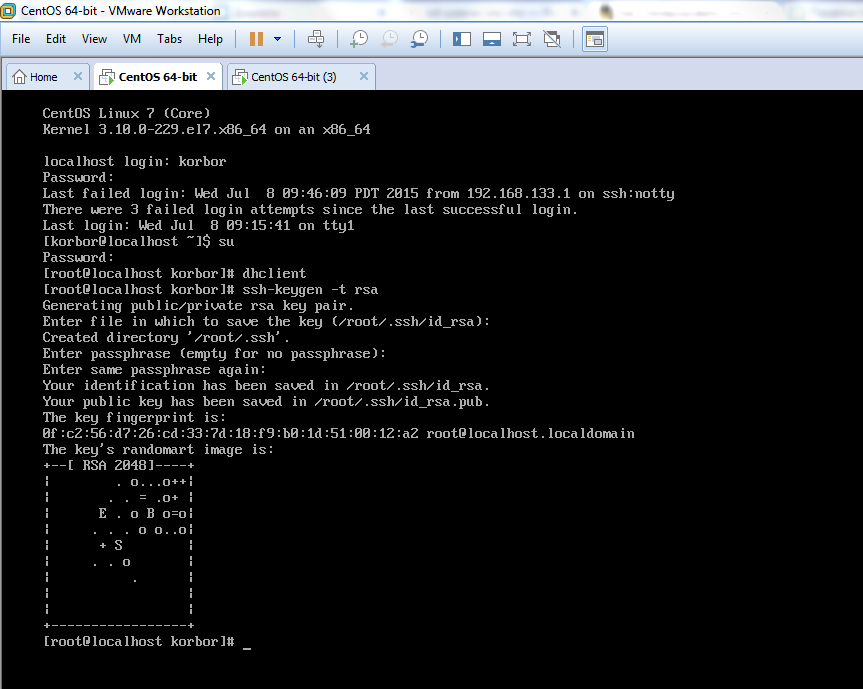
Когда у сервера SSH есть ваш открытый ключ в файле, и он видит, что вы запрашиваете соединение, он использует этот открытый ключ, чтобы создать и отправить вам т.н. вызов. Этот вызов является чем-то вроде зашифрованного сообщения, на которое должен поступить соответствующий ответ, чтобы сервер предоставил вам доступ. Безопасным это сообщение делает тот факт, что оно может быть прочитано только кем-то, у кого есть закрытый ключ. Открытый ключ может быть использован для зашифровки сообщения, но расшифровать то же самое сообщение он не сможет. Только вы, держатель закрытого ключа, будете иметь возможность корректно принять вызов и создать соответствующий ответ.

Этот этап вызов-ответ проходит незаметно для пользователя. До тех пор, пока у вас есть закрытый ключ, который обычно хранится в каталоге ~/.ssh/, ваш клиент SSH будет иметь возможность отправить правильный ответ серверу.

Поскольку закрытые ключи считаются конфиденциальной информацией, обычно они хранятся на диске в зашифрованном виде. По этой причине, когда запрашивается закрытый ключ, необходимо ввести пароль для расшифровки этого ключа. Внешне это может быть похоже на ввод пароля непосредственно на сервере SSH, но это не так: этот пароль используется только для расшифровки закрытого ключа в вашей локальной системе. Этот пароль не передается и не должен передаваться через сеть.

Для создания ключа вам необходимо ввести в консоли команду

***ssh-keygen -t rsa***



Далее везде нажимаем Enter, пропуская дополнительные вопросы о сохранении ключа в определенную папку и идентификационное слово.

Теперь ваши открытый и закрытый SSH-ключи должны быть сгенерированы. Откройте файловый менеджер и перейдите в директорию .ssh. Вы должны увидеть два файла: id\_rsa и id\_rsa.pub.

 Загрузите файл id\_rsa.pub в домашнюю директорию вашего удаленного хоста (предполагается, что удаленный хост работает под Linux). Подключитесь к удаленному хосту с помощью SSH и переместите открытый ключ в его целевую директорию с помощью команд:

***$ cat id\_rsa.pub >> ~/.ssh/authorized\_keys***

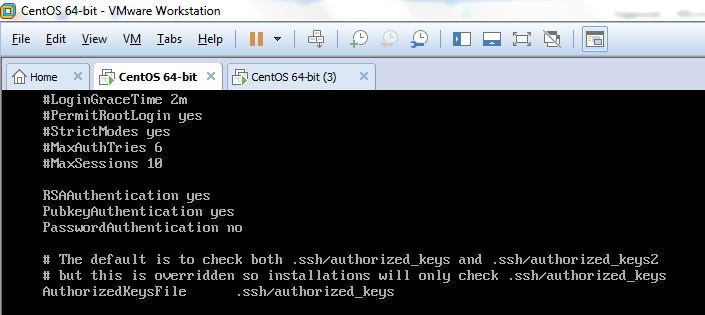
***$ rm id\_rsa.pub***

Убедитесь, что следующие атрибуты имеют корректные значения:

***RSAAuthentication yes***

***PubkeyAuthentication yes***

***PasswordAuthentication no***



Теперь вы можете соединяться со своим удаленным хостом с помощью команды:

***ssh -i /path-to-private-key korbor@198.168.133.128***

**6) ftp**

**FTP** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *File Transfer Protocol* — [протокол передачи](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%8B_%D0%BF%D0%B5%D1%80%D0%B5%D0%B4%D0%B0%D1%87%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) [файлов](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB)) — стандартный [протокол](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB), предназначенный для передачи файлов по TCP-сетям (например, Интернет). Использует 21й порт. FTP часто используется для загрузки сетевых страниц и других документов с частного устройства разработки на открытые [сервера](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%80%D0%B2%D0%B5%D1%80) [хостинга](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82%D0%B8%D0%BD%D0%B3).

Протокол построен на архитектуре «[клиент-сервер](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80)» и использует разные сетевые соединения для передачи команд и данных между клиентом и сервером. Пользователи FTP могут пройти аутентификацию, передавая логин и пароль[открытым текстом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%82%D0%B5%D0%BA%D1%81%D1%82), или же, если это разрешено на сервере, они могут подключиться анонимно. Можно использовать протокол [SSH](https://ru.wikipedia.org/wiki/SSH) для безопасной передачи, скрывающей (шифрующей) логин и пароль, а также шифрующей содержимое.

Первые клиентские FTP-приложения были интерактивными инструментами командной строки, реализующими стандартные команды и синтаксис. [Графические пользовательские интерфейсы](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) с тех пор были разработаны для многих используемых по сей день операционных систем. Среди этих интерфейсов как программы общего веб-дизайна вроде [Microsoft Expression Web](https://ru.wikipedia.org/wiki/Microsoft_Expression_Web), так и специализированные FTP-клиенты (например, FileZilla).

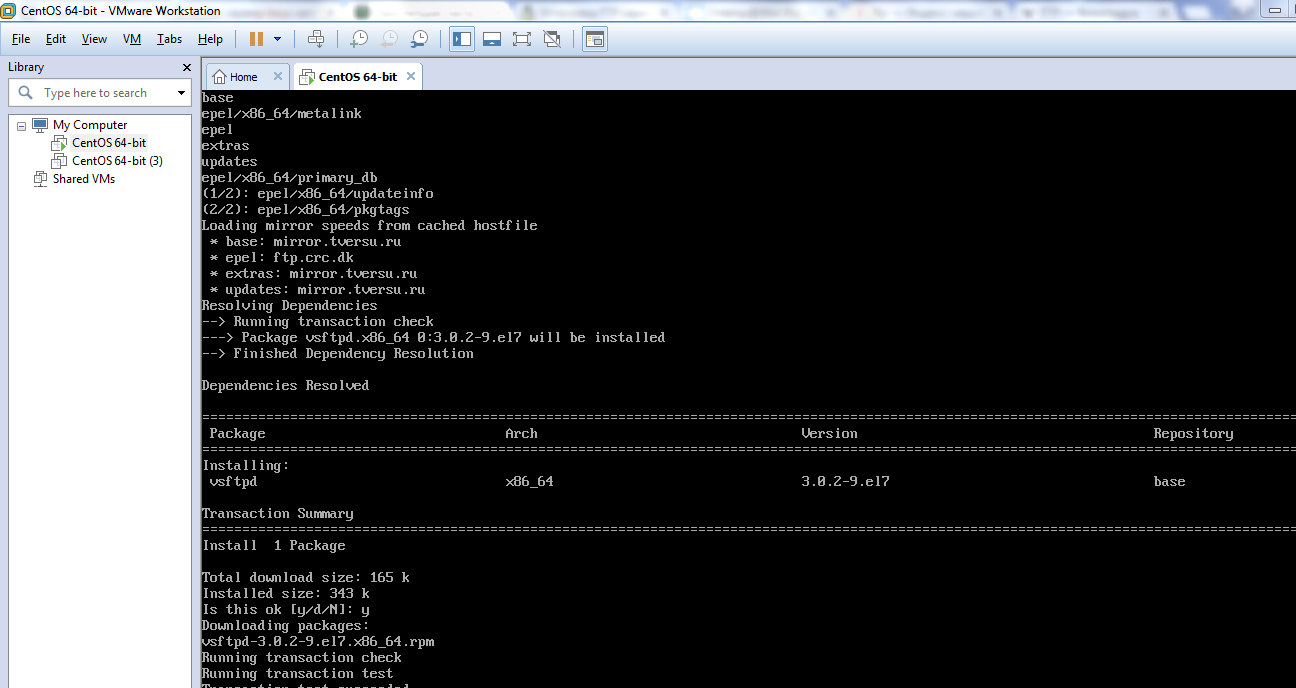
FTP является одним из старейших прикладных протоколов, появившимся задолго до [HTTP](https://ru.wikipedia.org/wiki/HTTP), и даже до [TCP/IP](https://ru.wikipedia.org/wiki/TCP/IP), в 1971 году. В первое время он работал поверх протокола [NCP](https://ru.wikipedia.org/wiki/Network_Control_Protocol)[[1]](https://ru.wikipedia.org/wiki/FTP#cite_note-1). Он и сегодня широко используется для распространения [ПО](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) и доступа к удалённым [хостам](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82).

Доступ к файлам на удаленном компьютере по протоколу FTP осуществляется с помощью программ, которые называются **FTP-клиентами** (в качестве примитивного FTP-клиента может использоваться www-браузер, например Opera, Firefox или Microsoft Internet Explorer). Практически все современные операционные системы включают также FTP-клиент для работы в командной строке, который так и называется «ftp».  
Если у вас есть интернет, то вы можете получить доступ к большому количеству информации, расположенной в различных уголках Сети. Для использования FTP необходим так называемый FTP-клиент, подключающийся к FTP-серверу (сервер, откуда скачиваются данные).Анонимный FTP позволяет подключаться к серверу даже не будучи на нем зарегистрированным (не имея на нем логина и пароля). Как правило, в качестве логина (имени пользователя) указывается anonymous, а в качестве пароля — ваш e-mail. Это делается на больших серверах для того, чтобы каждый мог скачать, к примеру, бесплатный дистрибутив Linux или какие-нибудь другие полезные программы.

FTP-клиент общается с FTP-сервером при помощи специальных FTP-команд (в зависимости от сервера они могут незначительно отличаться, но в целом набор команд более-менее стандартен). Тогда почему бы не дать возможность пользователю вводить эти команды, чтобы без посредника просматривать содержимое FTP-сервера, закачивать файлы, устанавливать режимы передачи. Именно так и было на заре интернета. Первопроходцы глобальной паутины торопливо набирали команды для FTP-сервера прямо из консоли. Существует такая возможность и поныне. Правда, если для пользователей Unix-систем такая манера общения с FTP весьма привычна, то обладатели Windows наверняка даже и не подозревают, что их система позволяет это делать. Надо отметить, что использовать консольный вариант FTP-клиента поначалу очень даже увлекательно, а в некоторых случаях и чрезвычайно полезно.

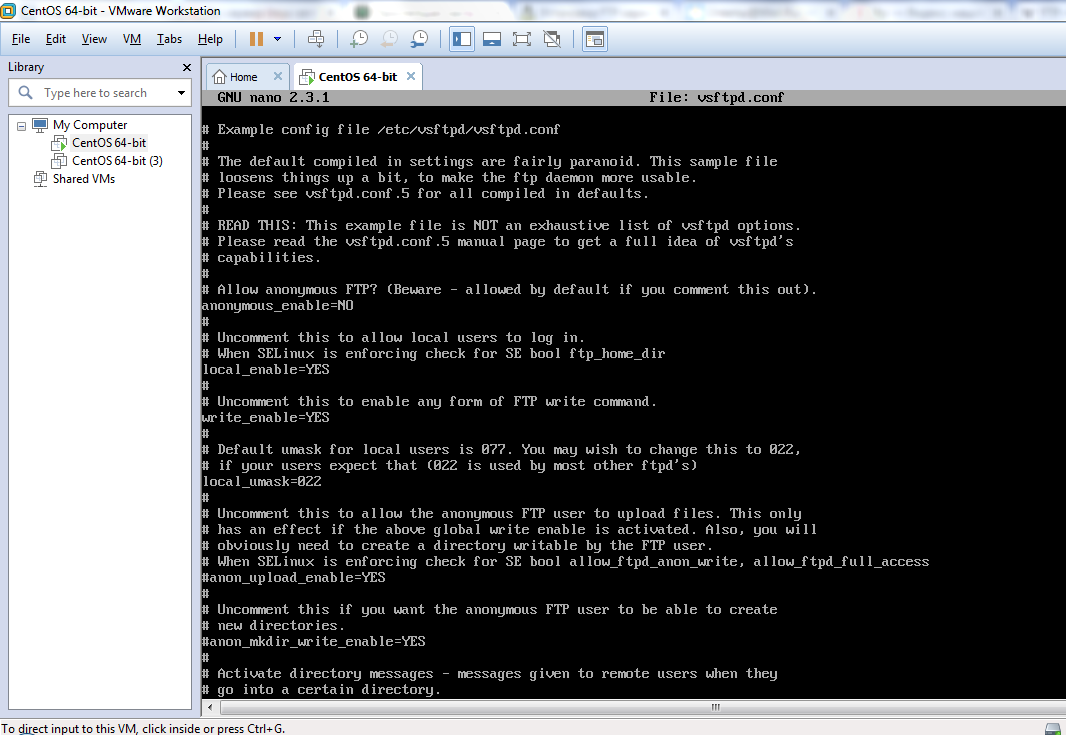
Установка ftp

***yum install vsftpd***

******

Откроем конфигурационный файл и поменяем настройки

***/etc/vsftpd/vsftpd.conf***



***anonymous\_enable=*NO**  
  
Запрещаем анонимным пользователям использовать наш сервер.

Запускаем наш сервис

***service vsftpd start***

Теперь, нам необходимо добавить пользователя который, собственно и будет использоваться для подключения к FTP.

***useradd FTPuser***

Задаем пароль для этого пользователя

***passwd FTPuser***

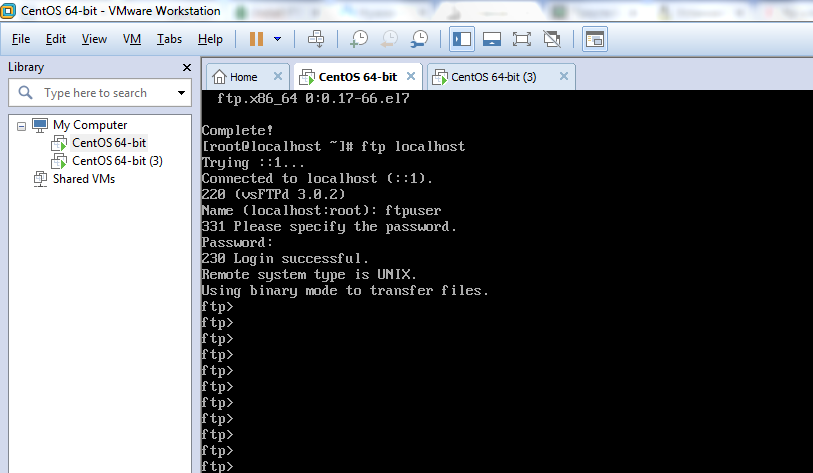
(вводим пароль)

скачиваем еще одну утилиту

***yum intstall ftp***

далее после установки вводим

***ftp localhost***



Доступ получен, можно передавать файлы